

Software Testing

Specifications

NoCap: Fact Checking for Articles using AI

Team Members:

Thomas Chamberlain tchamberlain2023@my.fit.edu

Varun Doddapaneni vdoddapaneni2023@my.fit.edu

Josh Pechan jpechan2023@my.fit.edu

Anthony Ciero aciero2022@my.fit.edu

Client and Faculty Advisor:

Dr. Marius Silaghi

Table of Contents

1. Introduction.....
2. Core Feature 1: Accepts URL/text as input.....
2.1 - Purpose.....
2.2 - Test Case: Paste plain text into input box.....
2.2.1 - Procedure.....
2.2.2 - Expected Output
2.2.3 - Edge cases
2.3 - Test Case: Submit by URL
2.3.1 - Procedure
2.3.2 - Expected Output
2.3.3 - Edge Cases
2.4 - Test Case: Multilingual articles
2.4.1 - Procedure
2.4.2 - Expected Output
3. Core Feature 2: Generates a Rating.....
3.1 - Purpose
3.2 - Test Case: Correctness on known inputs
3.2.1 - Procedure
3.2.2 - Expected Output
3.3 - Test Case: Performance
3.3.1 - Procedure
3.3.2 - Expected Output
3.4 - Test Case: Failure Behavior
3.4.1 - Procedure
3.4.2 - Expected Output
4. Core Feature 3: Rating explanation is provided.....
4.1 - Purpose
4.2 - Test Case: Explanation accuracy
4.2.1 - Procedure
4.2.2 - Expected Output
4.3 - Test Case: Explanation Format
4.3.1 - Procedure
4.3.2 - Expected Output
4.4 - Test Case: Evidence grounding
4.4.1 - Procedure
4.4.2 - Expected Output
4.5 - Test Case: Length/Reliability
4.5.1 - Procedure
4.5.2 - Expected Output

- 5. Core Feature 4: WCAG guidelines.....**
- 5.1 - Purpose
- 5.2 Test Case: Keyboard Navigation
 - 5.2.1 Procedure
 - 5.2.2 Expected Output
- 5.3 Test Case: Color Contrast
 - 5.3.1 Procedure
 - 5.3.2 Expected Output
- 5.4 Test Case: Text for Images
 - 5.4.1 Procedure
 - 5.4.2 Expected Output
- 5.5 Test Case: Screen Reader
 - 5.5.1 Procedure
 - 5.5.2 Expected Output

1. Introduction

This document defines the testing strategies for various project features and specific test cases for our website. The purpose of testing is to ensure that our system works reliably.

2. Core Feature 1: Accepts URL/text as input

2.1 - Purpose

To allow users to submit a URL or a direct text paste into our text box. The system must extract the article content, validate its input size, clean any data, and pass it to our rating engine.

2.2 - Test Case: Paste plain text into input box

2.2.1 - Procedure

- Navigate to our article submission website
- Paste a valid text article into the text area
- Click enter on keyboard/submit button near input box

2.2.2 - Expected Output

- UI accepts the text input and brings us to another part of the website that gives the rating
- Server responds with a rating and explanation for the rating

2.2.3 - Edge cases

- If the text pasted is too long, show an error message: “Text too large - maximum ... words”

2.3 - Test Case: Submit by URL

2.3.1 - Procedure

- Navigate to our article submission website
- Enter a valid URL into the URL text box
- Click enter on the keyboard or submit button near input box
- Observe whether the system extracts the main article text correctly in order

2.3.2 - Expected Output

- The system fetches the page, extracts the article text including: Title, Body, Author, and date if available
- If fetch fails, show an error message “Could not fetch page (404)”
- If the page blocks bots/subscription, show an error message “Content unavailable”

2.3.3 - Edge Cases

- For very long pages, the system extracts text and if it's over our limit, shows an error message “Article too long.”

2.4 - Test Case: Multilingual articles

2.4.1 - Procedure

- Navigate to our article submission website
- Submit a text paste or URL
- Click enter on the keyboard or submit button near input box
- Observe and make sure multilingual texts/articles are translated to English

2.4.2 - Expected Output

- The system translates text before the rating is outputted and the explanation is also translated to English.

3. Core Feature 2: Generates a rating

3.1 - Purpose

Compute and return a rating for the submitted article/text paste, following our defined rules. Ratings must be similarly reproducible and stable.

3.2 - Test Case: Correctness on known inputs

3.2.1 - Procedure

- Prepare a small batch (5) of articles with ratings assignment by humans following the same guidelines that the system will use
- Enter each article and record its rating

3.2.2 - Expected Output

- The system ratings should be within the acceptable range to pass
- Defects are reported as mismatches

3.3 - Test Case: Performance

3.3.1 - Procedure

- Submit a single article and measure end to end time
- Repeat across a number of samples (10)
- Run load test, 4 concurrent users submitting 500+ word articles at the same time

3.3.2 - Expected Output

- 90% of requests are completed under 30 seconds
- Under ~50 concurrent users, UI remains responsive and doesn't crash

3.4 - Test Case: Failure Behavior

3.4.1 - Procedure

- Simulate the rating engine downtime
- Submit an article

3.4.2 - Expected Output

- System retries, 2 attempts
- After both attempts and still fail, present an error message "Rating currently unavailable, please try later"

4. Core Feature 3: Rating explanation is provided

4.1 - Purpose

Produces a clear and accurate explanation that justifies the numeric rating. Explanation will reference article excerpts and grammar factors.

4.2 - Test Case: Explanation accuracy

4.2.1 - Procedure

1. Create different articles focusing on a single flaw
 - Good content but many grammatical errors
 - Good content but severely lacking in sources/evidence
 - Contains biased/opinionated language
2. Submit each and read explanation

4.2.2 - Expected Output

- Explanation generated points out the primary issue that was tested and cites a few snippets as examples in the explanation
- The rating score corresponds to the issues being tested

4.3 - Test Case: Explanation Format

4.3.1 - Procedure

- Submit a few articles

4.3.2 - Expected Output

- Every rating response includes a non empty string and includes at least one cited reason for the rating from the article
- UI displays the explanation on the same page near the numeric rating

4.4 - Test Case: Evidence grounding

4.4.1 - Procedure

- Submit an article with no author, cited sources, and external claims
- Check explanation for any references of facts that were not presented in the article

4.4.2 - Expected Output

- Explanations must not come up with outside facts or cite sources that were not originally present. Since our model uses external knowledge, any statements not originally in the article MUST be labeled as such.

4.5 - Test Case: Length/Readability

4.5.1 - Procedure

- Submit a long article (1,000+ words)
- Once an explanation is returned it should take into account UI considerations, Start with less than a 300 word explanation with a “Show More” button that will extend the page down to scroll until the full output is shown.

4.5.2 - Expected Output

- Explanation is truncated/limited by default to less than 300 words with a “show more” option
- Highlights remain accurate even when explanation is collapsed/expanded

5. Core Feature 4: WCAG Guidelines

5.1 - Purpose

Ensure that our website follows WCAG accessibility guidelines so that all users no matter their condition can use our website properly.

5.2 - Test Case: Keyboard Navigation

5.2.1 - Procedure

- Ensure all content of our website can be accessible using only the keyboard using keys such as tab, shift, enter, space, arrow keys

5.2.2 Expected Output

- All pages should be viewable and accessible using enter for searching, arrows for the navigation bar and more.

5.3 - Test Case: Color Contrast

5.2.1 Procedure

- Ensure the color combination choice is WCAG accessible by choosing colors that will be easily viewable on the color of background.

5.2.2 Expected Output

- WCAG color accessibility websites show which combination of text and background colors are unreadable, partly readable but not accessibility friendly, and accessible and readable to all users.

5.4 - Test Case: Text for Images

5.2.1 Procedure

- Add text to images that give a clear description of the image for accessibility use.

5.2.2 Expected Output

- Any image on our site will have a small caption; since most images will come from articles given to us or already in our database, we can use the article's captions to keep accuracy with the original article.

5.5 - Test Case: Screen Reader

5.2.1 Procedure

- Gives the option to read the screen aloud to allow vision impaired users another option of reading from our page.

5.2.2 Expected Output

- Allow the user to turn on/off screen reader, which will read the screen aloud.